Journal of NeuroEngineering and Rehabilitation

BioMed Central

Research

A novel asynchronous access method with binary interfaces

Jorge Silva*[1,4], Jorge Torres-Solis[1,2,3], Tom Chau[2,3] and Alex Mihailidis[4]

Address: [1]Komodo OpenLab, Toronto, Canada, [2]Bloorview Research Institute, Bloorview Kids Rehab, University of Toronto, Canada, [3]Institute of Biomaterials and Biomedical Engineering, University of Toronto, Canada and [4]Intelligent Assistive Technologies and Systems Lab, University of Toronto, Canada

Email: Jorge Silva* - jorge.silva@komodoopenlab.com; Jorge Torres-Solis - jorge.torressolis@utoronto.ca; Tom Chau - tom.chau@utoronto.ca; Alex Mihailidis - alex.mihailidis@utoronto.ca

* Corresponding author

Published: 29 October 2008

Received: 18 February 2008

Accepted: 29 October 2008

Journal of NeuroEngineering and Rehabilitation 2008, **5**:24 doi:10.1186/1743-0003-5-24

This article is available from: http://www.jneuroengrehab.com/content/5/1/24

Abstract

Background: Traditionally synchronous access strategies require users to comply with one or more time constraints in order to communicate intent with a binary human-machine interface (e.g., mechanical, gestural or neural switches). Asynchronous access methods are preferable, but have not been used with binary interfaces in the control of devices that require more than two commands to be successfully operated.

Methods: We present the mathematical development and evaluation of a novel asynchronous access method that may be used to translate sporadic activations of binary interfaces into distinct outcomes for the control of devices requiring an arbitrary number of commands to be controlled. With this method, users are required to activate their interfaces only when the device under control behaves erroneously. Then, a recursive algorithm, incorporating contextual assumptions relevant to all possible outcomes, is used to obtain an informed estimate of user intention. We evaluate this method by simulating a control task requiring a series of target commands to be tracked by a model user.

Results: When compared to a random selection, the proposed asynchronous access method offers a significant reduction in the number of interface activations required from the user.

Conclusion: This novel access method offers a variety of advantages over traditionally synchronous access strategies and may be adapted to a wide variety of contexts, with primary relevance to applications involving direct object manipulation.

Background

Many Disabled individuals require custom interfaces that enable them to access the devices they may wish to control. When appropriately designed, such interfaces take advantage of the user's known abilities, while eliminating reliance on onerous operational requirements. Thus, the design of appropriate user interfaces for Disabled individuals involves a process of understanding the needs, challenges and abilities of each user. In order to facilitate this process, it is necessary to count on widely available and highly adaptable tools that may be customized and combined in order to obtain the most appropriate solutions in each case. One such tool is the binary interface (commonly represented as a button or a switch), which, due to its simplicity and adaptability, has become a ubiquitous resource to overcome barriers to access for Disabled people.

A binary interface is formally defined as a device that may present only one of two distinct and stable states at any given time (e.g., on/off), which may be used to convey information between two entities [1]. Moreover, according to basic principles of information theory, binary interfaces are in fact the simplest possible means through which a user may communicate intent, since they represent the basic unit of information, namely, the binary digit or bit [2]. Therefore, binary interfaces may also be termed minimal interfaces. Minimal interfaces for Disabled users include other means of communication characterized by a low information storage (i.e., memory) capacity, this is the case, for example, with most brain-computer interfaces (BCI) currently available [3,4].

The problem of binary access

In order to communicate intent through a binary interface, a user must be able to intentionally determine, whenever necessary, which of the two possible states the interface should present. Thus, for example, in the case of a button, the user must be able to intentionally perform the mechanical actions required to press and release the button. Other binary interfaces may, for example, exploit the user's ability to produce a gesture [5] or blink [6] at will.

More recently, researchers have explored the detection of voluntary changes in physiological activity, such as brain [7] or electrodermal activity [8], in order to obtain a few distinct and repeatable patterns that, similarly to binary interfaces, may be used to communicate intent. These novel approaches may provide a means of access for those users whose intent may not be understood otherwise. Some of these physiological interfaces, although still minimal, are capable of respresenting more than 1 bit of information at once, however, due to a variety of design, measurement and contextual challenges, their implementation is generally simpler and more effective when only a binary mode of use is required.

In spite of all these advantages, binary interfaces also present significant limitations that preclude their use in a wide variety of access and control applications. Evidently, the binary nature of these interfaces makes them an ideal solution for the control of devices with intentional spaces that present only a dyad of possibilities (e.g., close-open, up-down, etc.) However, when access to more than two distinct outcomes is required for the successful control of a device, the limitations of the binary interface become immediately apparent. Figure 1 depicts this dilemma where a user is required to control a complex device by means of a binary interface.

Protocol-based binary access

Consider the set $S = \{s_0, s_1, s_2, \ldots, s_{\varsigma-1}\}$ containing all ς states available in a typical interface. Note that, with a binary interface, $\varsigma = 2$. This interface, is used to access the set $C = \{c_0, c_1, c_2, \ldots, c_{\kappa-1}\}$ of size κ, containing all possible outcomes available for selection. The initial limitation of the case $\kappa > \varsigma$ is typically overcome by the implementation of

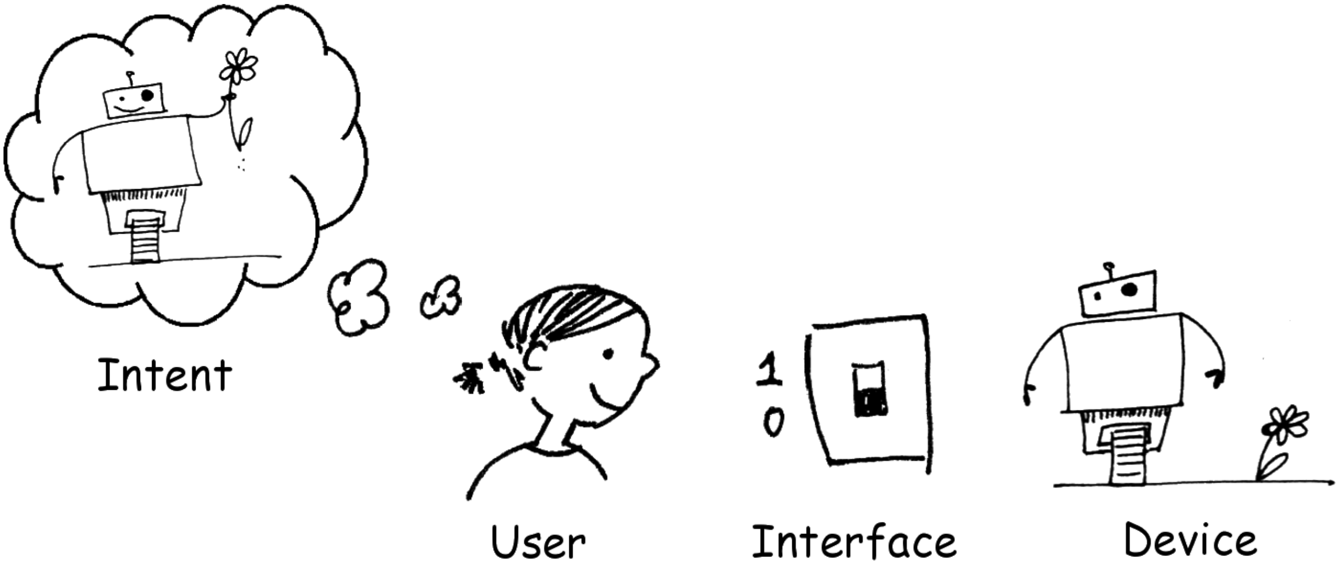


Figure 1
The problem of access with binary interfaces. A user is required to communicate intent, by means of a binary interface, to a device capable of more than two outcomes.

a time-bound protocol that enables the generation of a new set $S_T = \{f_0(t), f_1(t), f_2(t),...,f_{\kappa\text{-}1}(t)\}$ where each element $f_i(t) \in S_T$ is a time-dependent function composed by a unique sequence of channel states $s_i \in S$ with duration T . This time-based coding enables the direct mapping of each member $f_i(t)$, of the newly created set of functions S_T, to a unique message $c_i \in C$. Figure 2 shows two sample periodic state sequences $f_i(t)$ used to communicate messages through a binary interface (i.e., $\varsigma = 2$). The top sequence represents the hexadecimal number $9A_{HEX}$ as defined by the RS232 serial communication protocol. The bottom sequence represents the letter '*X*' as defined by the Morse code. Evidently, there are significant similarities between early electronic communication challenges and the use of binary interfaces by Disabled users. These similarities were quickly identified by interface designers who transferred the application of time-bound communication protocols to the implementation of access solutions for the Disabled. In fact, Morse-based communication and computer access methods are still being actively researched [9,10].

There are, however, some significant disadvantages with the use of time-bound protocols in the control of a device by a human operator. These stem mainly from the fact that both the transmitting and the receiving end must comply with the protocol used in the communication process. This requires users to either memorize all pairs $\{f_i(t), c_i\}$ mapping every device outcome $c_i \in C$ to its corresponding sequence $f_i(t) \in S_T$, or learn the time-coding rule $g(t) : f_i(t) \rightarrow c_i$ that may be used to generate the i-th sequence $f_i(t) \in S_T$ corresponding to the desired outcome

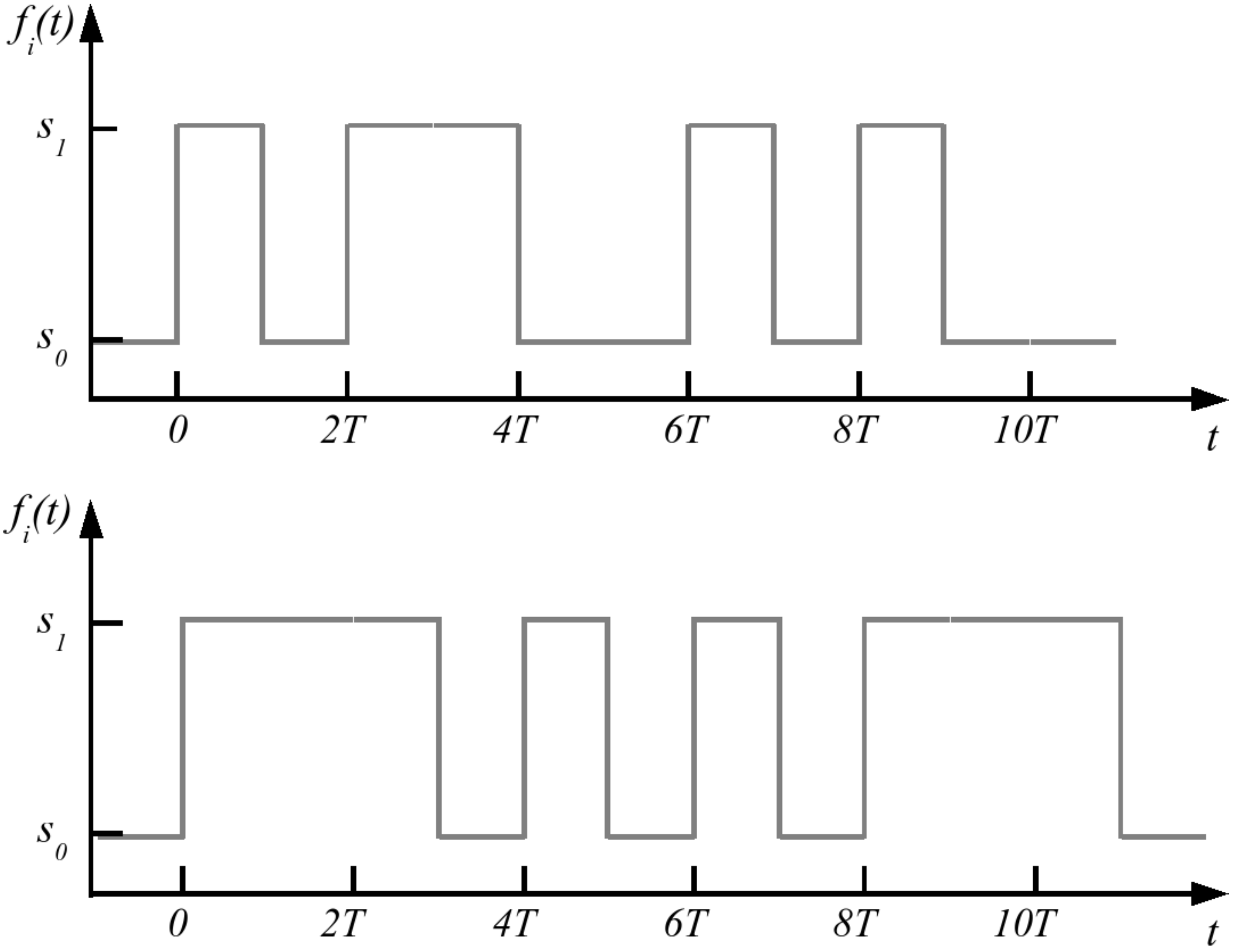


Figure 2
Sample state sequences f_i(t) used to communicate a particular message through a binary channel. The top trace represents the hexadecimal number $9A_{16}$ in the RS232 serial communication protocol. The bottom trace represents the letter '*X*' in Morse code.

$c_i \in C$. Evidently, depending on individual abilities, this requirement will affect different users to varying degrees. However, the number κ of device outcomes that can be made available to the user will be largely limited by the user's memory capacity as well as the complexity of the protocol. Therefore, this requirement will impose, in all cases, an upper boundary κ_{max} on κ (i.e., $\kappa \leq \kappa_{max}$).

Scanning-based binary access

In order to maximize the value of κ_{max}, feedback systems with varying degrees of complexity have also been developed. Some of these are designed to remind the user of the protocol's guidelines [11], while others, relying on periodic sensory cues, may completely eliminate the need for memorization [12]. This latter category includes all scanning access methods, commonly used by Disabled people nowadays. With scanning methods, all possible outcomes are presented to the user, at once, by means of a sensory pathway (usually visual and/or auditory). During operation, the outcomes are automatically highlighted, one by one, at a given rate according to the user's abilities. In order to indicate intent, users are required to activate the binary interface whenever their desired outcome is highlighted. This process results in the generation of time-dependent sequences $f_i(t)$ similar to the ones depicted in Figure 2. However, in contrast to the protocols formerly described, there is far more tolerance for variance in the period T during which the state of the interface must be maintained. Furthermore, because scanning methods rely mostly on the feedback information about the state of the scanning process presented to the user, there are usually sequences $f_i(t) \in S_T$ that correspond to more than one outcome $c_i \in C$. These characteristics make scanning methods accessible to a wider variety of users and extend the range of potential applications beyond those available with the more formal protocols described above. However, scanning access methods still present a significant drawback: the timing of the interaction is controlled by an automatic agent, not by the user. Thus, even after the user has already decided on the intended outcome, (s)he must still wait until this outcome is highlighted by the automated scanning process in order to communicate the intention. A variety of strategies have been proposed to optimize this process and therefore reduce the time required for the intended outcome to be selected [12,13], however, the basic principle remains the same. As a result, with scanning access methods, it is time, rather than memory capacity or protocol complexity, that limits the maximum number, κ_{max}, of device outcomes that can be made accessible to the user.

Synchronous vs. asynchronous binary access

Because of the external time constraints imposed on the user, both protocol-based and scanning-based access methods are more generally defined as *synchronous* in the study of human machine interfaces (HMI). Within this field, a synchronous access strategy may be defined as a method that requires users to comply with one or more time constraints in order to communicate intent with a minimal interface. This implies that, with synchronous access strategies, there will always be an additional delay in the process of selection of the intended outcome.

Conversely, *asynchronous* access methods do not place any time constraints on the users. Thus, users may initiate control of the device at any time without having to wait for external cues. Furthermore, no protocols are necessary because a single interface activation is sufficient to transmit a full unambiguous message to the device under control. Therefore, there is no additional delay in the selection of the intended outcome. When using binary interfaces, this is easily achievable when the intention space only presents two possibilities. That is, when the number of possible device outcomes is $\kappa = 2$.

Consider, for example, a wall switch with states $S = \{s_0 : \text{UP}, s_1 : \text{DOWN}\}$ used to select one of the two possible outcomes $C = \{c_0 : \text{ON}, c_1 : \text{OFF}\}$ of a light bulb. In this case, it is possible to map directly each outcome $c_i \in C$ with a particular interface state $s_i \in S$ in order to establish a suitable control strategy:

$$c_i = \begin{cases} c_0 : \text{ON} & \text{if } s_0 : \text{UP} \\ c_1 : \text{OFF} & \text{if } s_1 : \text{DOWN} \end{cases} \tag{1}$$

According to Equation (1), every time the position of the wall switch changes, the behavior of the light bulb will change accordingly. Thus, a single change in the wall switch represents a full, unambiguous command sent to the light bulb, allowing the latter to respond immediately.

It has always been assumed that this kind of asynchronous access is impossible in cases where the number κ of outcomes C required to control a device is greater than the number ς of states S available in the interface. However, the method presented in this paper may be used with minimal interfaces presenting as few as $\varsigma = 2$ stable states, in order to access, asynchronously, sets of device outcomes of any size $\kappa \in \{2, 3, 4,...\}$. This includes those belonging to analog, as well as multidimensional domains, such as the movement parameters of an object in a 3-dimensional space. As a result, a variety of activities not typically available to Disabled users, may now be made accessible to them.

In the following sections, we provide details on the mathematical development of the proposed method for asynchronous access, the necessary guidelines for its implementation, and an initial evaluation based on a simulated control task. Our concluding remarks and suggestions for future work, are summarized in latter sections.

A new method for asynchronous binary access

To present the proposed method for asynchronous access, we will initially focus on the case where a binary interface must be used to access a set of outcomes of arbitrary size, in order to control a particular device or perform a specific task. It is important to note that this analysis was originally prompted by the solution of a specific access challenge, namely, the development of an appropriate strategy to facilitate *binary navigation control*. In the context of disability engineering, binary navigation control consists of enabling users to voluntarily define and/or modify the motion parameters of an object in space, at any time, by means of a binary interface. Binary navigation control is thus required to enable most activities involving object manipulation with binary interfaces (e.g., single-switch drawing). Many such activities are currently inaccessible to binary and other minimal interface users. For example, when defining suitable alternatives for computer access, Shein (1997) described single-switch, computer-aided drawing as an exceptionally challenging activity that, unlike many other computer-related tasks, may not be broken into predictable sequences accessible through standard synchronous methods [14].

Consider a user who attempts to employ a single button (single-switch) to access a device requiring a set C of $\kappa > 2$ outcomes. The button, in turn, presents only $\varsigma = 2$ possible states $S = \{s_0 : \text{released}, s_1 : \text{pressed}\}$. Thus, a simple mapping strategy such as the one shown in Equation (1) may not be used.

Initially, we may define the transition from state s_0 to state s_1 (i.e., a button press) as an intentional, user-prompted change in the interface. We will call this event . For the sake of simplicity, we will assume that the opposite transition (i.e., a button release) is not an intentional event and thus, will not represent a change in the interface.

According to the principle of asynchronous access described above, every time occurs, the behavior of the device must be changed. In other words, a new device outcome $c \in C$ must be selected. Note that this principle suggests that the event is only necessary when the behavior of the device is unacceptable to the user since this would be the only instance where a change in the behavior of the device would be welcome. Conversely, if the behavior of the device is already consistent with the user's intention, the event is not required. In other words, in our example, the button should be used to indicate the presence of unacceptable behaviors (i.e., errors) in the device through the intentional generation of events .

Let n be the count of consecutive events , and $c_{[n]} \in C$ the device outcome chosen in response to the n-th occurrence of . The fundamental principle of asynchronous access may then be simply defined as:

$$c_{[n]} \neq c_{[n\text{-}1]} \tag{2}$$

This principle states that when the n-th event occurs, the resulting device outcome $c_{[n]}$ must be different from the outcome $c_{[n\text{-}1]}$ preceding it. We call this principle a negative acknowledgement (NAK) signaling process because the user is required to activate the interface only when the device behaves erroneously. This term has been borrowed from the analogous error detection, out-of-band, signaling system for error control, often used in telecommunications [15], which, because of its simplicity, has been shown to reduce the communication costs (in terms of time and bandwidth) in environments with significant processing constraints [16].

The exclusion mask

With the exception of Equation (2), there is no additional information that could help us determine, precisely, which of the remaining elements $c \neq c_{[n\text{-}1]}$ of C should be selected as the outcome $c_{[n]}$. The top trace in Figure 3 shows an alternative graphical representation of this knowledge, which may be formally defined as

$$\mathcal{X}_{[n]}(c) = \begin{cases} 1 & \text{if } c = c_{[n-1]} \\ 0 & \text{if } c \neq c_{[n-1]} \end{cases} \tag{3}$$

Here, the element $c_{[n\text{-}1]}$ is assigned a maximum value of $\mathcal{X}_{[n]}\,(c = c_{[n\text{-}1]}) = 1$. This value represents an absolute certainty that $c_{[n\text{-}1]}$ should be excluded from the selection of the device behavior $c_{[n]}$ as stated in Equation (2). Conversely, all other elements share the minimum value $\mathcal{X}_{[n]}\,(c \neq c_{[n\text{-}1]}) = 0$, which represents absolute uncertainty about their possibility of exclusion from the selection of $c_{[n]}$. Thus, $\mathcal{X}_{[n]}\,(c)$, which may only take values in the range [0, 1], constitutes a numerical representation of the certainty of exclusion of a given outcome $c \in C$ from the selection of $c_{[n]}$. In other words, $\mathcal{X}_{[n]}\,(c)$ may be used to describe a range of assumptions (from weak $\mathcal{X}_{[n]}\,(c) \simeq 0$ to strong $\mathcal{X}_{[n]}\,(c) \simeq 1$) regarding the unsuitability of outcomes in the choice $c_{[n]}$. This function will be termed the *spatial exclusion mask* of $c_{[n]}$.

The representation of the NAK principle in Equation (2) by means of the spatial exclusion mask $\mathcal{X}_{[n]}\,(c)$ may ini-

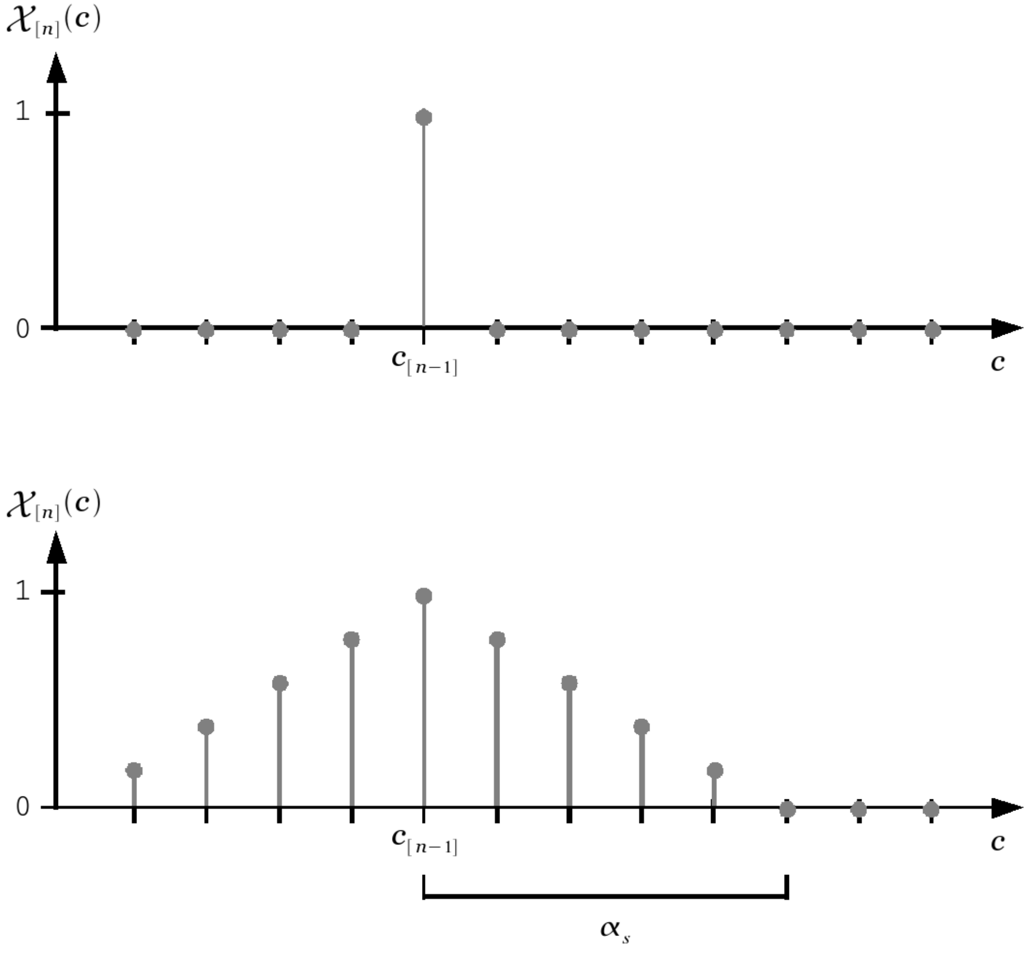


Figure 3 $\mathcal{X}_{[n]}$

Sample spatial exclusion masks $\mathcal{X}_{[n]}$ (c). The top mask represents the fundamental knowledge implied by the principle of asynchronous access. Because of its maximum value $\mathcal{X}_{[n]}$ ($c = c_{[n-1]}$) = 1, the element $c_{[n-1]}$ cannot be chosen when selecting a new device outcome $c_{[n]}$. The bottom mask represents the assumption that outcomes similar to $c_{[n-1]}$ should also be excluded from the selection of $c_{[n]}$.

tially seem unnecessary. However, as it will be demonstrated in the following sections, this mask introduces a framework for the numerical representation of contextual knowledge that may be used to optimize the choice $c_{[n]}$ in response to a single binary event .

Spatial assumptions

In any typical access problem, it is expected that the set of outcomes required to control a device may be numerically arranged in a domain where the distance between similar outcomes is shorter than the distance between dissimilar ones. In that case, outcomes in the neighborhood of $c_{[n-1]}$ would be expected to resemble $c_{[n-1]}$. This expectation has an important implication in the definition of the spatial exclusion mask $\mathcal{X}_{[n]}(c)$ because it suggests that all those outcomes near (i.e., similar to) $c_{[n-1]}$ should also be given high (i.e., $\mathcal{X}_{[n]}(c) \simeq 1$) values of exclusion from the selection of the outcome $c_{[n]}$. This is because it may be assumed that outcomes in the neighborhood of $c_{[n-1]}$ are too similar to $c_{[n-1]}$ to cause a significant change in the behavior of the device. This assumption, however, is not as certain as the fundamental principle in Equation (2), because it is not directly implied by the event . Moreover, the certainty of this assumption should be lower for outcomes that are far apart from $c_{[n-1]}$ than for outcomes that are closer to $c_{[n-1]}$.

Thus, a suitable spatial exclusion mask $\mathcal{X}_{[n]}(c)$ representing these assumptions may be:

$$\mathcal{X}_{[n]}(c) = \begin{cases} 1 - \dfrac{r}{\alpha_S} & \text{if } r \le \alpha_s \\ 0 & \text{if } r > \alpha_s \end{cases} \qquad (4)$$

where $r = |c - c_{[n-1]}|$ is the distance between a given outcome $c \in C$ and the outcome $c_{[n-1]} \in C$ preceding the n-th event . In turn, α_s is a positive integer used to define the support boundaries $c_{[n-1]} \pm \alpha_s$ of $\mathcal{X}_{[n]}(c)$. The bottom trace in Figure 3 depicts the updated spatial exclusion mask defined in Equation (4). Note that in the limit $\alpha_s \to 0$, Equation (4) will become Equation (3) as depicted by the top trace in Figure 3.

Evidently, the introduction of the exclusion mask $\mathcal{X}_{[n]}(c)$ suggests that the best choice of $c_{[n]}$ will be the element $c \in C$ that minimizes $\mathcal{X}_{[n]}(c)$ (i.e., $c_{[n]} = \text{argmin } \mathcal{X}_{[n]}(c)$). In both cases presented (Figure 3), there is more than one element c that fulfills this condition, thus, the selection of $c_{[n]}$ is still ambiguous. However, note that the updated mask $\mathcal{X}_{[n]}(c)$ described in Equation (4) reduces the number of eligible outcomes $c \in C$ to those that lie beyond the support limits $c_{[n-1]} \pm \alpha_s$ of the spatial exclusion mask. In fact, if α_s is large enough, a unique solution may be found. The significance of this reduction in the number of eligible outcomes for the choice $c_{[n]}$ will be evident in later discussions. In the meanwhile, note that any function $\mathcal{X}_{[n]}(c)$ with support limits $c_{[n-1]} \pm \alpha_s$ that decreases monotonically from $c_{[n-1]}$ to $c_{[n-1]} \pm \alpha_s$, may be used to represent the spatial assumptions described above.

Temporal assumptions

The spatial exclusion mask $\mathcal{X}_{[n]}(c)$ in Equation (4) represents a series of assumptions, with varying degrees of certainty, that outcomes in the spatial neighborhood of $c_{[n-1]}$ should not be eligible in the selection of the subsequent device behavior $c_{[n]}$. Similarly, these assumptions may be extended, starting with the outcome $c_{[n-1]}$, back in time throughout the past history $\{c_{[n-2]}, c_{[n-3]}, c_{[n-4]},...\}$ of selected outcomes. Thus, as in the spatial case, outcomes in the temporal neighborhood of $c_{[n-1]}$ (i.e., immediately preceding $c_{[n-1]}$) should also share a high value of exclusion from the choice $c_{[n]}$, while outcomes that belong to the remote past history of $c_{[n-1]}$ should be assigned lower values. This is because we may assume that if the recently chosen outcome $c_{[n-1]}$ has already been excluded, there is a high level of certainty that this outcome will not be desired in the near future. However, over time, this outcome should be made available. Evidently, extending this assumption through time requires a memory process that enables the storage of historical information on all outcomes preceding the n-th event . This information must then be available at the time $t_{[n]}$, when this event occurs, in order to inform the selection of $c_{[n]}$. The spatial exclusion mask $\mathcal{X}_{[n]}(c)$, introduced above, cannot be employed for this purpose since it only describes assumptions valid at $t_{[n]}$ without providing any means to describe assumptions associated with the set of past events $\{n\text{-}1, n\text{-}2, n\text{-}3,...\}$. Thus, an additional mechanism that enables the incorporation of historical information in the choice $c_{[n]}$ becomes necessary.

The exclusion estimate

Consider the function $\Upsilon(c, t)$ depicted in the discrete time sequence presented in Figure 4. This function describes the viscoelastic deformation of the 1-dimensional domain composed of all elements $c \in C$. The figure shows parallel bands representing the state of the domain at regular time intervals. In order to elucidate the progression of time, the bands have been colored from dark to clear corresponding to the transition from older to more recent states of the domain. We will assume $\Upsilon(c, t)$ has been left undisturbed for a long time $t < t_{[n]}$allowing it to maintain its natural at state (i.e., $\Upsilon(c, t < t_{[n]}) = 0$ for all possible outcomes $c \in C$). Then, at a given time $t = t_{[n]}$, the domain is subject to a deformation $\mathcal{X}_{[n]}(c)$ as depicted by the bottom trace in Figure 3. By definition, the viscoelastic deformation process would allow $\Upsilon(c, t)$ to recover its natural state. However, as depicted by subsequent bands, this will only happen gradually over time.

The sequence in Figure 4 depicts the temporal memory effect inherent to the mechanical property of viscoelasticity. Note that this property fulfills the requirements stated in the previous section for the incorporation of temporal assumptions in the choice $c_{[n]}$. In particular, in the context of asynchronous access, the deformation $\Upsilon(c, t)$ subject to

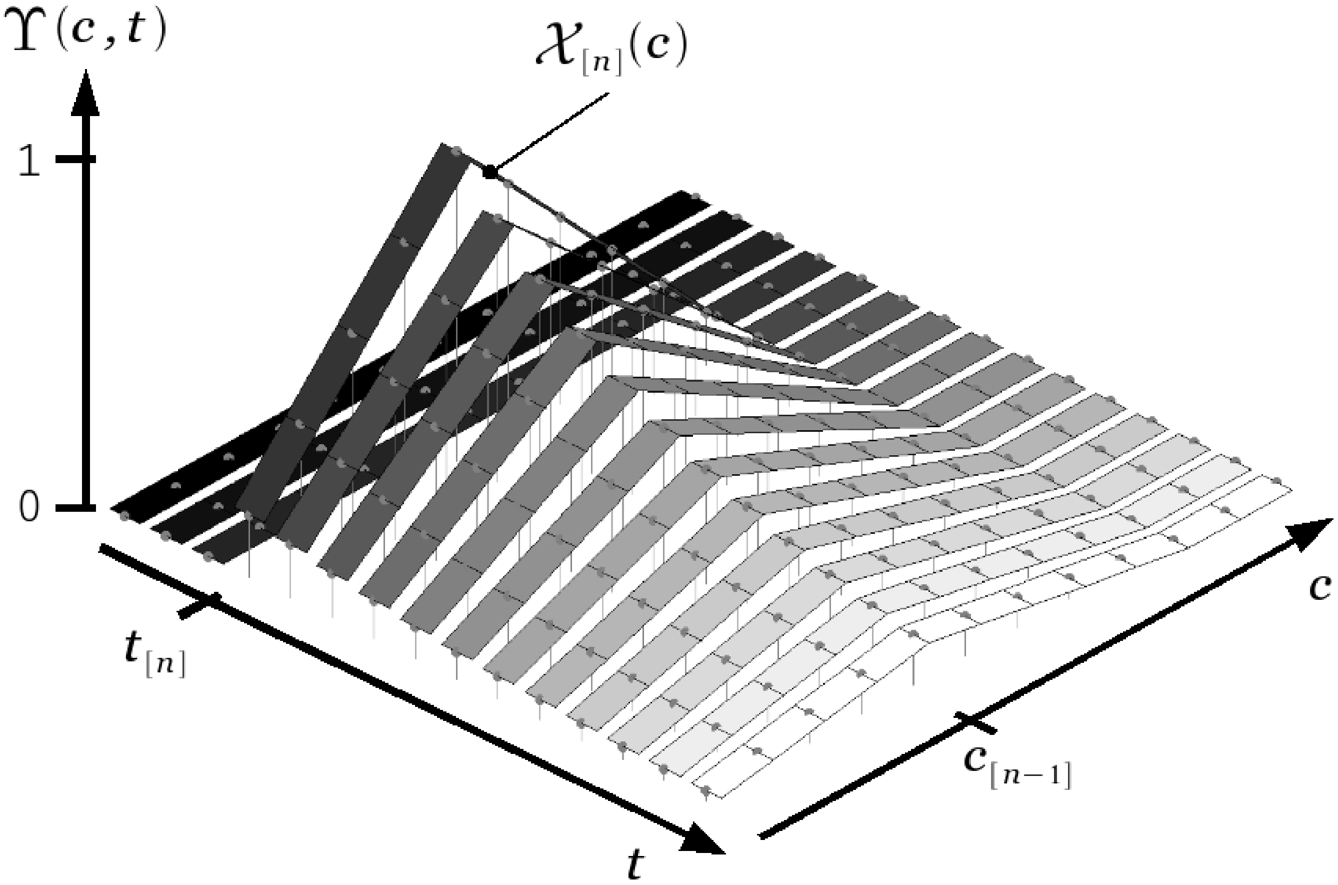


Figure 4

A discrete time sequence of the viscoelastic deformation of the 1-dimensional domain *C* depicted by function Υ(*c*, *t*). The deformation $\mathcal{X}_{[n]}(c)$, occurring at time $t_{[n]}$, experiences a steady decay over time.

consecutive disturbances $\mathcal{X}_{[n]}(c)$, would allow recent assumptions $\mathcal{X}_{[n]}(c)$ (i.e., those in the temporal neighborhood of $c_{[n-1]}$) to be assigned higher values of exclusion than former ones. In other words, the function $\Upsilon(c, t)$ may be used to record the the full set of historical assumptions represented by all spatial exclusion masks $\{\mathcal{X}_{[n-1]}(c), \mathcal{X}_{[n-2]}(c), \mathcal{X}_{[n-3]}(c) \ldots\}$ preceding the n-th event . A simple recursive algorithm may be used to represent this process.

Let Δt be the period between the time $t_{[n]}$ of the n-th event and the time $t_{[n-1]}$ of the preceding event, that is

$$\Delta t = t_{[n]} - t_{[n-1]} \tag{5}$$

and $\Upsilon_{[n]}(c)$ the function $\Upsilon(c, t)$ evaluated at time $t_{[n]}$, that is

$$\Upsilon_{[n]}(c) = \Upsilon(c, t = t_{[n]}) \tag{6}$$

The spatial and temporal assumptions previously introduced may then be represented, simultaneously, as the occurrence of disturbances $\mathcal{X}_{[n]}(c)$ on $\Upsilon_{[n]}(c)$ with viscoelastic decay $\mathcal{H}_{[n]}(\Delta t)$

$$\Upsilon_{[n]}(c) = \mathcal{H}_{[n]}(\Delta t)\Upsilon_{[n-1]}(c) + \mathcal{X}_{[n]}(c)\{1 - \mathcal{H}_{[n]}(\Delta t)\Upsilon_{[n-1]}(c)\} \tag{7}$$

We will refer to $\Upsilon_{[n]}(c)$ in Equation (7) as the *exclusion estimate* of the current choice $c_{[n]}$. Note that, $\Upsilon_{[n]}(c)$ is defined recursively in terms of the exclusion estimate $\Upsilon_{[n-1]}(c)$ of the previous choice $c_{[n-1]}$. All functions Υ, $\mathcal{X}$ and $\mathcal{H}$ are constrained to the range [0, 1]. The function $\mathcal{H}_{[n]}(\Delta t)$, used to apply a viscoelastic decay on $\Upsilon_{[n-1]}(c)$, should decrease monotonically with increasing values of Δt. A suitable choice for $\mathcal{H}_{[n]}(\Delta(t)$ may thus be the family of functions

$$\mathcal{H}_{[n]}(\Delta t) = e^{-\Delta t/\tau} \tag{8}$$

where τ is a time constant always greater than zero. The exponential decay described in Equation (8) derives from the behavior of real viscoelastic systems such as the discharge of an electric capacitor or the restoration of a mechanical shock absorber [17]. In all these cases, the constant τ is termed the *viscoelastic constant* and it is directly proportional to the duration of the viscoelastic restoration of $\Upsilon_{[n]}(c)$.

Note that $\mathcal{X}_{[n]}(c)$ and $\mathcal{H}_{[n]}(\Delta t)$ are weighting functions acting on the spatial and temporal domains, respectively, of the exclusion estimate $\Upsilon_{[n]}(c)$. While the spatial exclusion mask $\mathcal{X}_{[n]}(c)$ ensures that outcomes similar to $c_{[n-1]}$ are excluded from the choice $c_{[n]}$, the function $\mathcal{H}_{[n]}(\Delta t)$ ensures that recent exclusion estimates $\Upsilon(c, t)$ are remembered while old ones are forgotten. Thus, $\mathcal{H}_{[n]}(\Delta t)$ is our *temporal exclusion mask*. Note that the support of $\mathcal{H}_{[n]}(\Delta t)$ is defined for values in the range $[0, \alpha_t]$ with $\alpha_t > 0$. In the case of the family of functions in Equation (8), $\alpha_t = \infty$.

The definition of the exclusion estimate $\Upsilon_{[n]}(c)$ in Equation (7), which now integrates spatial and temporal assumptions, suggests that the best possible choice of $c_{[n]}$ should be the element $c \in C$ that minimizes $\Upsilon_{[n]}(c)$. Thus,

$$C_{[n]} = \text{argmin } \Upsilon_{[n]}(c) \tag{9}$$

Figure 5 shows a discrete time sequence of the evolution of $\Upsilon(c, t)$, according to Equation (9), where three different events occur at consecutive times. Note that it has taken only two events , with corresponding exclusion masks

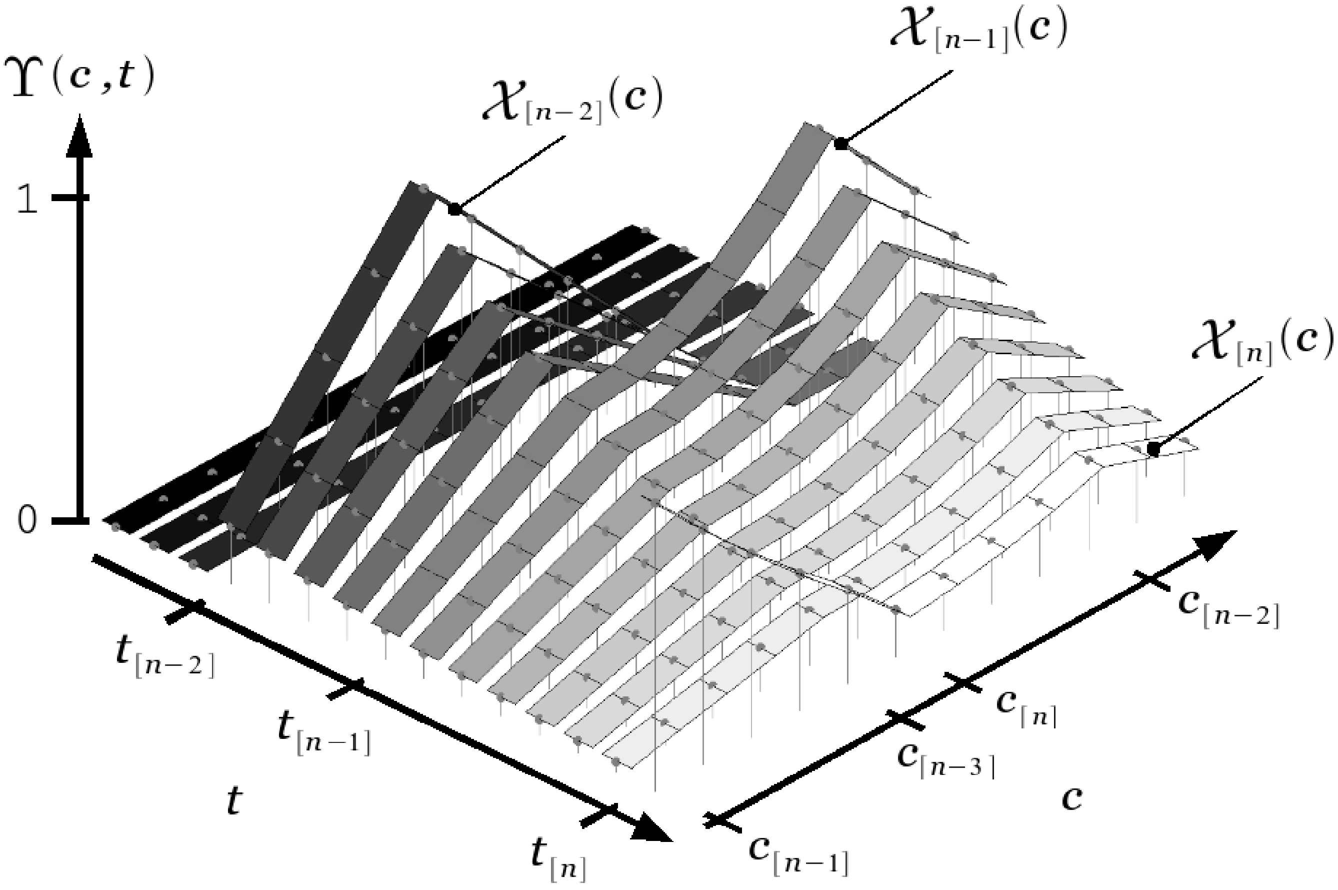


Figure 5
Discrete time sequence of the evolution of the exclusion estimate Υ(*c*, *t*) according to Equation (9). Three consecutive events are presented at different intervals.

$\mathcal{X}_{[n-2]}(c)$ and $\mathcal{X}_{[n-1]}(c)$, for $\Upsilon(c, t)$ to converge from a state of absolute uncertainty for $t < t_{[n-2]}$, to a unique solution $c_{[n-1]}$ at $t = t_{[n-1]}$.

Equation (9) summarizes the decision process proposed for the asynchronous selection of a new device outcome $c_{[n]} \in C$ in response to a single binary event , consisting, in our example of single-switch access, of an intentional button press. Thanks to the assumptions incorporated in $\mathcal{X}_{[n]}(c)$ and $\mathcal{H}_{[n]}(\Delta t)$, the number of eligible device outcomes in the choice $c_{[n]}$ is significantly reduced. In fact, with the appropriate parameters, Equation (9) will consistently converge to a unique solution soon after the interaction between the user and the device under control is initiated.

The process for asynchronous access presented here incorporates a number of desirable properties that make it easy to implement and adaptable to a wide variety of contexts. Among these properties are:

- There are no restrictions on the time at which a particular event may occur. For users, this translates into the ability to respond immediately to a change in their intentions or an unexpected external disturbance on the device under control.

- The recursive nature of the exclusion estimate $\Upsilon_{[n]}(c)$ eliminates the need for the implicit calculation of the effects of the set of historical assumptions $\{\mathcal{X}_{[n-1]}(c), \mathcal{X}_{[n-2]}(c), \mathcal{X}_{[n-3]}(c), \ldots\}$ on the selection of $c_{[n]}$, thus reducing the processing power and memory storage capacity required for the implementation of the proposed method for asynchronous access.

- There is no limit on the number κ of outcomes C that may be made available to the user through this method. In fact, the set C may be defined as a continuous interval of all possible real valued outcomes $c \in [c_{min}, c_{max}]$, where c_{min} and c_{max} are the lower and upper boundaries of C, respectively. Evidently, in this case, $\kappa = \infty$.

Summary

1. According to Equation (2), when the n-th event occurs, the device outcome $c_{[n]}$ must be different from the outcome $c_{[n-1]}$ immediately preceding it. In other words, there is absolute certainty that $c_{[n-1]}$ should be excluded from the selection of $c_{[n]}$. Thus, the event , which represents a voluntary, user-prompted change in the interface, should be employed by users as an error indicator. This requires users to generate events every time the behavior of the device is inconsistent with their intentions.

2. Even though the exclusion principle in Equation (2) is the only knowledge implied, with absolute certainty, by the occurrence of event , it is also possible to assume, although with a lower degree of certainty, that behaviors similar to $c_{[n-1]}$ should also be excluded from the selection of $c_{[n]}$. This assumption is defined by the spatial exclusion mask $\mathcal{X}_{[n]}(c)$, a function with values in the range [0, 1] and support $c_{[n-1]} \pm \alpha_s$, decreasing monotonically from $\mathcal{X}_{[n]}(c = c_{[n-1]}) = 1$ (i.e., the strongest assumption of exclusion) to $\mathcal{X}_{[n]}(c = c_{[n-1]} \pm \alpha_s)$ (i.e., the weakest assumption of exclusion) as candidate outcomes $c \in C$ become decreasingly similar to $c_{[n-1]}$.

3. It may also be assumed that device outcomes resulting from recent selections (i.e., immediately preceding the n-th event), should be excluded from the selection of $c_{[n]}$, while outcomes that belong to the remote past of n should become eligible. The incorporation of this assumption is made possible through the introduction of the exclusion estimate $\Upsilon_{[n]}(c)$ and the temporal exclusion mask $\mathcal{H}_{[n]}(\Delta t)$, where Δt is, according to Equation (5), the period between the time $t_{[n]}$ of the n-th event and the time $t_{[n-1]}$ of its predecessor. According to Equation (7), the exclusion estimate $\Upsilon_{[n]}(c)$, which is recursively defined in terms of the exclusion estimate $\Upsilon_{[n-1]}(c)$ of the preceding event, acts as a viscoelastic domain storing the set of historical deformations $\{\mathcal{X}_{[n]}(c), \mathcal{X}_{[n-1]}(c), \mathcal{X}_{[n-2]}(c), \ldots\}$ subject to a viscoelastic decay described by the temporal exclusion mask $\mathcal{H}_{[n]}(\Delta t)$. Thus, $\mathcal{H}_{[n]}(\Delta t)$ must decrease monotonically from $\mathcal{X}_{[n]}(\Delta t = 0) = 1$ to $\mathcal{H}_{[n]}(\Delta t = \infty) = 0$. Note that the functions $\mathcal{X}_{[n]}(c)$ and $\mathcal{H}_{[n]}(\Delta t)$ act as weighting masks on $\Upsilon_{[n]}(c)$ updating the certainty of exclusion, from the choice $c_{[n]}$, for every candidate outcome $c \in C$, according to reasonable spatial and temporal assumptions, respectively.

4. Once the exclusion estimate $\Upsilon_{[n]}(c)$ is calculated, it will be possible to make an informed decision regarding the best possible choice of $c_{[n]} \in C$ according to Equation (9).

Implementing the proposed method

In order to successfully implement the method for asynchronous binary access presented above, some additional considerations are required.

Initialization

Note that the decision process described in Equation (9) does not specify the characteristics of the exclusion estimate $\Upsilon_{[0]}(c)$ before the first (i.e., $n = 1$) event is generated by the user. In fact, there is no information regarding the value of the initial device outcome $c_{[0]}$ either, and without this knowledge, the recursive process described in Equation (7) may not be initialized. Thus, even before the user initiates interaction with the device, a virtual selection $c_{[0]}$ must be made. Similarly to the case where the concept of viscoelasticity was first introduced, we may assume that before the first user-prompted event (i.e., $t < t_{[1]}$) the exclusion estimate had been left undisturbed for a long time, thus allowing it to recover its natural flat state. That is, $\Upsilon_{[0]}(c) = 0$ for all outcomes $c_{[0]} = c \in C$. Moreover, since all values $c \in C$ fulfill the condition in Equation (9), we would then be obliged to draw $c_{[0]}$ from a uniform distribution of C. Consequently, this random selection of $c_{[0]}$ may be used to initialize the decision process Equation (9). Note that it is not necessary to communicate the virtual choice $c_{[0]}$ to the device under control. Thus, the device may remain undisturbed until after the first user-prompted event occurs. In this case, the virtual choice $c_{[0]}$ will only be used to obtain the first exclusion mask $\mathcal{X}_{[1]}(c)$ at $t_{[1]}$, enabling the calculation of the estimate $\Upsilon_{[1]}(c)$. The resulting outcome $c_{[1]}$ will then be the first to affect the device's behavior. From the perspective of the user, it will appear that the outcome $c_{[1]}$ has been drawn randomly from a uniform distribution. However, as explained here, this is only the case for the virtual choice $c_{[0]}$, since, according to Equation (9) $c_{[1]}$ will be drawn from a more restricted distribution where a subset of the elements $c \in C$ (i.e., $\sim c_{[0]} \pm \alpha_s$) have already been excluded.

An alternative (and, in fact, more useful) procedure consists of initializing $\Upsilon_{[0]}(c)$ with random white noise in the interval of real numbers [0, 1]. This minimizes the probability of having multiple candidates for the virtual choice $c_{[0]}$, since it is expected that, after this initialization, $\Upsilon_{[0]}(c)$ will present a unique minimum value, which may then constitute the virtual choice $c_{[0]}$. The advantage of this method over the one initially proposed, resides in the fact that with the latter, $\Upsilon_{[n]}(c)$ will more likely converge to a unique solution from the beginning (i.e., $n = 1$) of the interaction. In fact, this also allows for the prediction of future selections of $c_{[n]} \in C$ with a significant degree of confidence.

Anchorage

If the viscoelastic constant τ is too long, or a significant number of events occur in a short amount of time, the exclusion estimate $\Upsilon_{[n]}(c)$ may accumulate constant offsets from previous, but still remembered deformations $\mathcal{X}(c)$. Due to the discretization process inherent to any numerical implementation of the proposed method (e.g., on a computer), this offset accumulation may in fact cause saturation of the exclusion mask $\Upsilon_{[n]}(c)$. That is, $\Upsilon_{[n]}(c) \simeq 1$ for all outcomes $c \in C$. If saturation occurs, the information storage capacity of the exclusion estimate will be completely eliminated, thus, preventing the selection of reasonable outcomes $c_{[n]}$ derived from the spatial and temporal assumptions introduced before.

In order to prevent the occurrence of saturation, constant offsets must be eliminated at all times from the exclusion estimate $\Upsilon_{[n]}(c)$. This may be achieved by subtracting the value of $\Upsilon_{[n]}(c = c_{[n]})$ from the function $\Upsilon_{[n]}(c)$. That is

$$\Upsilon_{[n]}(c) \Leftarrow \Upsilon_{[n]}(c) - \Upsilon_{[n]}(c = c_{[n]}) \qquad (10)$$

where $\Upsilon_{[n]}(c = c_{[n]})$ is the value of $\Upsilon_{[n]}(c)$ evaluated at the recently obtained outcome $c_{[n]}$. Evidently, if $\Upsilon_{[n]}(c = c_{[n]})$ is already zero, Equation (10) will have no effect on $\Upsilon_{[n]}(c)$. This process of elimination of the offset of the exclusion estimate $\Upsilon_{[n]}(c)$ is termed *anchorage*. The process of anchorage has no effect on the decision $c_{[n]}$, since this decision only depends on the relative exclusion value of a given outcome $c \in C$ as compared to the rest of the elements of C.

Algorithm

The following list summarizes the sequential steps required for the implementation of the proposed asynchronous access method.

1. Originally, nothing is known about the intention of the user regarding the behavior of the device. Thus, the exclusion estimate $\Upsilon_{[0]}(c)$ may be initialized with white noise in the range [0, 1]. This results in the definition of the virtual choice $c_{[0]}$ and the exclusion mask $\mathcal{X}_{[1]}(c)$, which precede any user interaction and, therefore, any change in the behavior of the device.

2. When the n-th intentional binary event occurs, the period Δt is calculated according to Equation (5) and used to obtain the decay $\mathcal{H}_{[n]}(\Delta t)$ through a suitable function such as Equation (8). Subsequently, the intention estimate $\Upsilon_{[n]}(c)$ is updated according to Equation (7).

3. The corresponding n-th device outcome $c_{[n]}$ may now be obtained according to Equation (9). This outcome is immediately transmitted to the device which experiences a change in behavior.

4. The exclusion estimate $\Upsilon_{[n]}(c)$ is anchored according to Equation (10).

5. The exclusion mask $\mathcal{X}_{[n]}(c)$ is updated to $\mathcal{X}_{[n+1]}(c)$ through a suitable function such as Equation (4).

6. For subsequent events , the process is repeated from (ii) above.

In addition, the fundamental spatial and temporal assumptions require their corresponding exclusion masks $\mathcal{X}_{[n]}(c)$ and $\mathcal{H}_{[n]}(\Delta t)$ to have the following properties:

- The spatial exclusion mask $\mathcal{X}_{[n]}(c)$ must decrease monotonically from $\mathcal{X}_{[n]}(c = c_{[n-1]}) = 1$ to $\mathcal{X}_{[n]}(c = c_{[n-1]} \pm \alpha_s) = 0$. The support of this function will be defined in the range $[c_{[n-1]} - \alpha_s, c_{[n-1]} + \alpha_s]$.

- The temporal exclusion mask $\mathcal{H}_{[n]}(\Delta t)$ must decrease monotonically from $\mathcal{H}_{[n]}(\Delta t = 0) = 1$ to $\mathcal{H}_{[n]}(\Delta t = \alpha_t) = 0$. The support of this function will be defined in the range $[0, \alpha_t]$ where $\alpha_t > 0$.

Note that although these assumptions are reasonable given the access problem proposed, there is no limit to the number and/or kind of assumptions that may be incorporated into $\mathcal{X}_{[n]}(c)$ and $\mathcal{H}_{[n]}(\Delta t)$. For example, one could deliberately exclude a particular outcome $c_i \in C$ (i.e., $\mathcal{H}_{[n]}(c_i) = 1$) or all events occurring before a certain memory threshold Δt_0 (i.e., $\mathcal{H}_{[n]}(\Delta t < \Delta t_0) = 0$) in response to some contextual knowledge.

Evaluation

We have presented a method for asynchronous binary access based on the selection of a particular outcome $c_{[n]}$ that may be immediately transmitted to the device under control in response to the single n-th binary event . However, we have not yet given any consideration to the case when the selected outcome $c_{[n]} \in C$ is inconsistent with the user's intention. This is, in fact, a very likely possibility if we consider that, according to the NAK signaling process previously described, by generating the event the user is simply requesting a change in the behavior of the device. However, there are no means to specify which of the outcomes $c \in C$ will be the most appropriate. Thus, if the outcome $c_{[n]} \in C$ chosen after the n-th event is unacceptable, the user will be required to generate another event hoping to obtain the desired outcome with the subsequent choice $c_{[n+1]} \in C$. Users will be required to repeat this process until the behavior of the device is consistent with their intention.

For the typical binary interface user, generating the event will require some kind of effort. Thus, measuring the number of events required to reach a particular target outcome $c_\gamma \in C$ would provide a benchmark for the evaluation of the cost associated with the proposed method. Note, however, that this measure arises from a naturally uncertain (i.e., stochastic) process and thus, may only be described in terms of probability.

Let N be the number of intentional binary events required to reach a series of typical target outcomes $c_\gamma \in C$, it is possible to measure the fraction $P(N \leq X)$ of targets c_γ that will require X or less events to be reached. This is known in probability theory as the cumulative distribution function (CDF) of the random variable N [18].

Figure 6 depicts a sample CDF corresponding to two different processes of selection of a specific outcome c, drawn from a uniformly distributed set C of size $\kappa = 8$, in response to consecutive binary events . Both processes follow a geometric distribution with CDF defined as

$$P(N \leq X) = 1 - (1 - p)^X \tag{11}$$

where p is the probability of making a correct choice for any given attempt.

The lower trace in Figure 6 (i.e., $p = 0.125$) results from a selection with substitution where all outcomes $c \in C$ are eligible on every trial n. Conversely, the trace with $p = 0.143$ results from a selection without substitution where the outcome $c_{[n-1]}$ is eliminated from the n-th trial. This reduction in the number of eligible outcomes in the choice $c_{[n]}$ increases the probability p of making a correct choice at any given trial n. Thus, as indicated by the dashed lines, with the latter process of selection without substitution, there is a marginal gain in the fraction $P(N \leq$

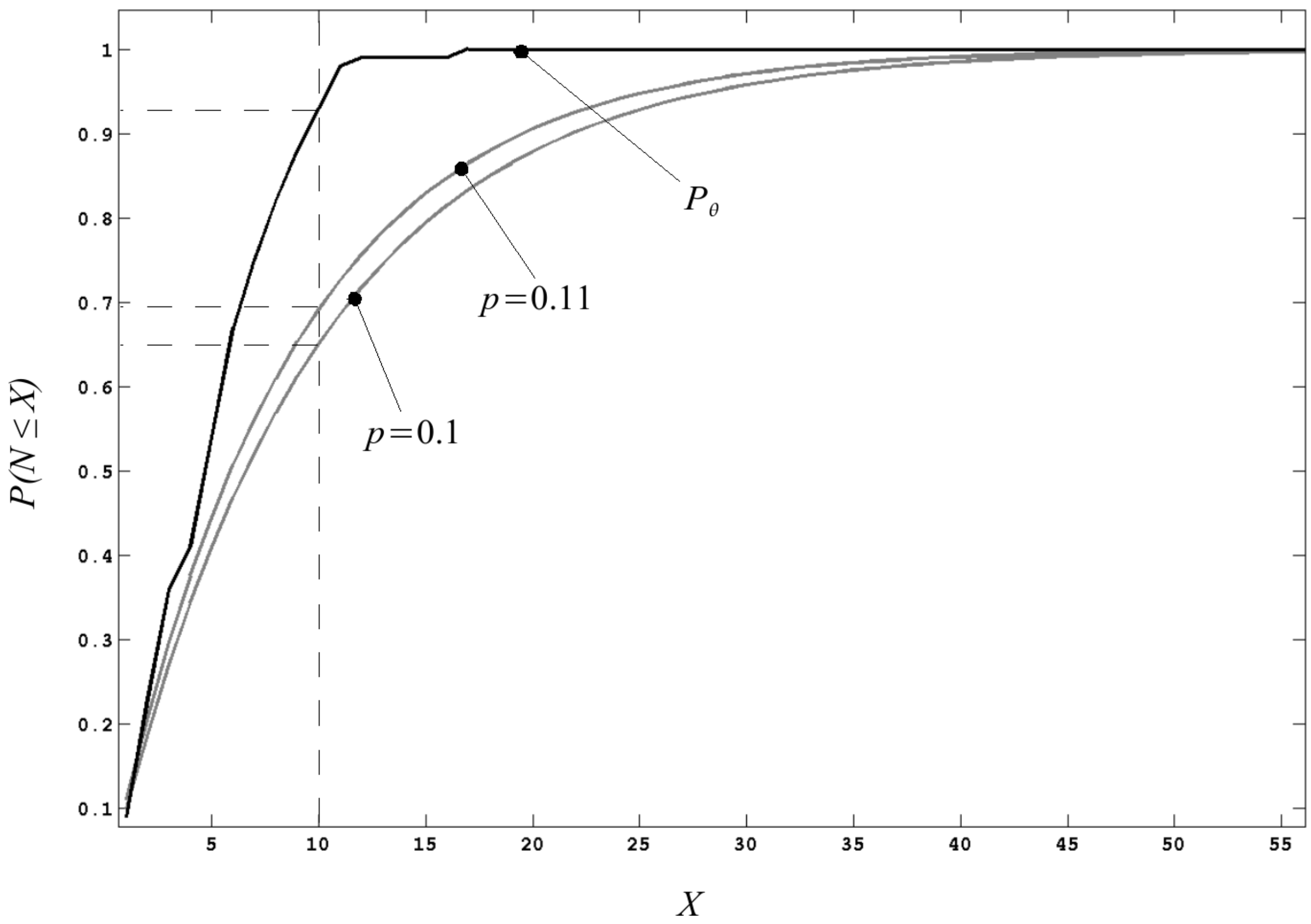


Figure 6
Cumulative distribution functions obtained from the process of selection of a given outcome c ∈ C subject to a fractional tolerance σ = 10% of the full range of C. The lower gray trace corresponds to a selection with substitution with probability of success p = 0.1. The middle gray trace corresponds to a selection without substitution p = 0.11 and the upper black trace corresponds to the proposed method for asynchronous access with parameters ϖ = 0.05, τ = 5}.

X) of targets c_γ that may be reached with X = 10 or less trials. The process of selection without substitution described above, is identical to the fundamental principle of asynchronous access presented in Equation (2), which describes the selection of a new device outcome $c_{[n]}$ in response to the user-prompted event . Thus, as demonstrated in Figure 6, the incorporation of the knowledge implied by this principle, translates directly into a reduction of the cost associated with the use of the device (i.e., a reduction in the expected number, N, of events required to reach a target outcome c_γ). Similarly, it would be desirable to evaluate the impact that the additional assumptions $\mathcal{X}_{[n]}(c)$ and $\mathcal{H}_{[n]}(\Delta t)$, incorporated in the exclusion estimate $\Upsilon_{[n]}(c)$, may have on the cost of use of the device. However, the CDF associated with these assumptions is not easy to obtain since it depends on a variety of parameters specific to the particular context of application. Thus, in order to evaluate the performance of the proposed access method, we completed a series of simulations for a select case of device control by a binary interface user.

Methods

In order to determine the expected impact on the cost of use of a device associated with the proposed method for asynchronous binary access, a Monte Carlo simulation [19] of a simple access task was performed. In this simulated environment, a computer model of a typical user was implemented. This model user was then required to employ a single binary interface in order to select a series of predefined targets c_γ from a set C of κ = 100 outcomes required to control a device (e.g. the volume of a TV). It was assumed that the main mode of monitoring the behavior of the device by the model user was visual. Thus, as soon as the model user 'observed' that the behavior of the device was inconsistent with the required target, an intentional event would be generated. A total of 1000 elements c_γ were initially drawn randomly with replacement from C in order to establish the predefined sequence of target outcomes that the user was required to select in order to successfully control the device. It is important to note that real access applications are likely to involve sequences of correlated actions rather than independent ones. Thus, our choice of random, uncorrelated targets represents an extreme case likely to constitute a lower boundary of performance for the proposed access method.

Each target in the sequence was presented to the model user until it was reached. Then, the next target in the sequence was presented, and so on. The objective of this simulation was to measure the number N of intentional events that would be required from the user in order to reach each target c_γ. This process was repeated 6 times for a total of 6000 targets per trial. This number was sufficient to quantify the statistical nature of N and obtain an estimate of its CDF for each case evaluated.

Modelling the user

As mentioned before, it was assumed that the model user was able to monitor the behavior of the device under control through visual means. This process would involve a series of delays as a result of the time required by the user to process the visual information and, if necessary, generate the event . Thus, in order to obtain an accurate model of the visual reaction time t_r, which includes both the visual perception and motor reaction times, an initial experiment was performed with a real user. During this experiment, the real user was requested to respond to simple visual stimuli presented on a computer screen. The stimuli consisted of the appearance of a white circle on a black background after random delays of 1 to 3 seconds. The user was instructed to press a button (defined as the event) immediately after the stimulus (i.e., the white circle) appeared on the screen. The experiment was performed using the open source software package PXLab, which can be used to accurately measure the user's reaction time t_r defined as the period from the presentation of the stimulus, to the generation of the intentional event . A histogram of the reaction times, t_r, was obtained with a total of 100 trials. This histogram was used to represent the model user in the Monte Carlo simulations introduced above. Thus, for each event , a reaction time, t_r, was randomly drawn from the histogram. The expected value $\overline{t_r}$ of this user's reaction time was ~213 ms, which is consistent with previous research on the topic [20]. Thus, it may be assumed that the statistical model, represented by the histogram obtained, was an accurate estimate of user behavior incorporating the stochastic nature of the interaction between a real user and a device.

Cases for evaluation

According to the proposed method of asynchronous control, there are few restrictions to the definitions of the exclusion masks $\mathcal{X}_{[n]}(c)$ and $\mathcal{H}_{[n]}(\Delta t)$. As a result, there is an infinite number of functions that comply with the basic requirements of both of these functions. We will focus on the evaluation of a single family of functions for each of the exclusion masks defined. These functions have already been introduced and correspond to some of the simpler sets of assumptions that may be made in compliance with the necessary requirements for the linear spatial exclusion mask $\mathcal{X}_{[n]}(c)$ in Equation (4) and the exponential temporal exclusion mask $\mathcal{H}_{[n]}(\Delta t)$ in Equation (8).

Each case for evaluation was defined by a set, $\underset{\sim}{\theta}$, of three parameters:

- The tolerance σ of the choice $c_{[n]}$. This parameter was used to define a boundary around the target outcome c_γ within which the choice $c_{[n]}$ was considered acceptable. In other words, the target outcome c_γ was reached if

$$\left|\frac{c[n]-c_\gamma}{c_{max}-c_{min}}\right| < \frac{\sigma}{2} \qquad (12)$$

- The width $\omega = \alpha_s \cdot (c_{max} - c_{min})^{-1}$ of the linear spatial exclusion mask $\mathcal{X}_{[n]}(c)$. This parameter specified the fraction of the full length of C defining the support boundaries of $\mathcal{X}_{[n]}(c)$ as defined in Equation (4).

- The viscoelastic constant τ of the temporal exclusion mask $\mathcal{H}_{[n]}(\Delta t)$ as defined in Equation (8). This parameter defined the expected size, in seconds, of the memory window of the exclusion estimate $\Upsilon_{[n]}(c)$.

Table 1 shows the admissible and selected test values for all parameters in $\underset{\sim}{\theta}$. In the context of this simulation, σ represented a requirement of the particular device under control while ω and τ characterized the algorithms employed to enable access to the device. In other words,

Table 1: Admissible and test values for the parameter set $\underset{\sim}{\theta}$ evaluated

Parameter	Admissible Values	Test Values
σ	[0, 1]	{0.05, 0.1, 0.15, 0.20}
ω	[0, ∞]	[0.02, 0.2] in regular increments of 0.01
τ	[0, ∞]	$\tau = e^z$ where z ∈ [-2.5, 3.75] in regular increments of 0.25

the simulation evaluated the performance of a number of algorithms described by ω and τ on the solution of particular access problems with a σ requirement. The test values for the parameter set $\underset{\sim}{\theta}$ were selected according to generalized but expected contexts of human-machine interaction, within relatively broad intervals. Thereore, this simulation may only be used to characterize the general properties of the proposed method for asynchronous access. In order to evaluate the performance of this method in specific applications, more complete simulations incorporating the appropriate parameters are necessary.

In total, 1976 sets $\underset{\sim}{\theta} = \{\sigma, \omega, \tau\}$ were evaluated, and, as mentioned before, each set consisted of 6000 separate trials where the model user was requested to reach a specific target c_γ using a single binary interface and the proposed method for asynchronous access.

Performance measure

According to Figure 6, for every possible value of X, the value of the fraction $P(N \le X)$ will be higher for algorithms that reduce the cost of access when compared to a simple random selection with substitution. Furthermore, the higher the value $P(N \le X)$, the more cost-effective the associated algorithm will be. Therefore, if the differences between the corresponding values $P(N \le X)$ for a given algorithm and a simple random selection are accumulated over X, it will be possible to obtain an overall score for each of the cases evaluated. We defined the overall score $\Gamma(\underset{\sim}{\theta})$ as

$$\Gamma(\theta) = \sum_{X=1}^{\infty} P_{\underset{\sim}{\theta}}(N \le X) - \sum_{X=1}^{\infty} 1-(1-\sigma)^{X} \qquad (13)$$

where $P_{\underset{\sim}{\theta}}(N \le X)$ is the CDF obtained for a particular set $\underset{\sim}{\theta} = \{\sigma, \omega, \tau\}$. Note that Γ is a relative measure of performance with reference to a process of random selection with substitution subject to the given tolerance σ . This latter process is captured by the second term in Equation (13) and defined in Equation (11). In the limit $X \to \infty$, both terms of the subtraction tend to 1 thus, in practice, it is only necessary to consider a sufficiently large number for X. For example, in the case presented in Figure 6, the value $X = 40$ would be an appropriate limit for the summations.

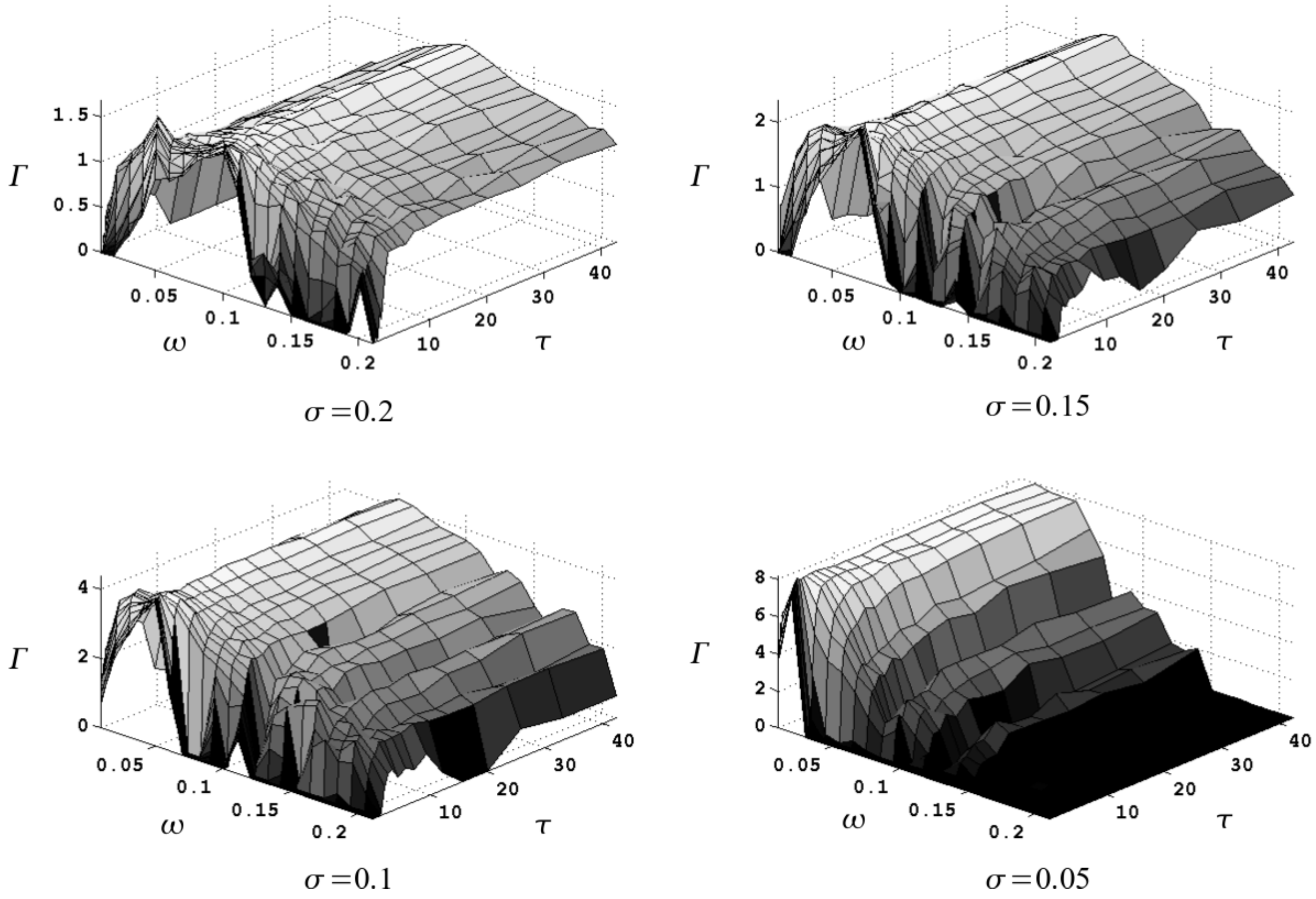
According to Equation (13), positive values of $\Gamma(\underset{\sim}{\theta})$ indicate lower usage costs. Thus, in order to optimize the cost, $\Gamma(\underset{\sim}{\theta})$ must be maximized. Conversely, negative values of $\Gamma(\underset{\sim}{\theta})$ would indicate a disastrous performance (i.e., even worse than a random guess). Finally, a value $\Gamma(\underset{\sim}{\theta}) = 0$ would indicate similar performance between a random guess and the proposed algorithm with the particular parameter set $\underset{\sim}{\theta}$. However, in such cases, the additional complexity of the proposed method would not justify its application. Thus, these sets should also be avoided.

Results and discussion

Figure 6 presents the CDF resulting from a single case $\underset{\sim}{\theta} = \{\sigma = 0.1, \omega = 0.05, \tau = 5\}$ as compared to the CDF obtained from a simple selection from a uniform distribution, with ($p = 0.1$) and without ($p = 0.11$) substitution, subject to the same tolerance requirement $\sigma = 0.1$. Note that, for a given value $X = 10$, the proposed method provides an advantage in excess of 20% in the fraction $P_{\underset{\sim}{\theta}}(N \le X)$ of trials completed.

Furthermore, this method reaches a maximum value $P_{\underset{\sim}{\theta}}(N \le X) = 1$ with less than half the events required by the other selection processes. In other words, the proposed method demands over 50% less effort from the user. With the parameter values specified above, the value of relative performance as defined in Equation (13) was $\Gamma(\underset{\sim}{\theta}) \approx 4.25$

Figure 7 shows all 1976 cases evaluated. Cases with relative performance values lower than zero were plotted here as $\Gamma(\underset{\sim}{\theta}) = 0$. Note that only a small fraction of cases exhibited this unacceptable value of performance.



$\sigma = 0.2$ $\sigma = 0.15$

$\sigma = 0.1$ $\sigma = 0.05$

Figure 7

Relative performance Γ for all sets of parameters $\underset{\sim}{\theta}$ = {σ, ω, τ} evaluated. The viscoelastic constant τ has a negligible effect beyond ~10 $\overline{t_r}$. Furthermore, with lower values of tolerance σ, the set of acceptable parameters {ω, τ} is reduced. However, the relative gain Γ may be significantly larger than that obtained with greater tolerances.

As mentioned before, the results of this simulated experiment must be interpreted with care. In particular, we have identified three significant concerns i) a real application is likely to involve other cognitive processes in addition to the simple visual reaction time used here to model the user, ii) the control of a real device is likely to involve a series of correlated targets instead of the independent ones proposed in our experiment, and iii) users can fail trying to activate the interface and cause a delay, but, worst, the user can involuntarily activate it even if (s)he is happy with the current choice.

Regarding the first concern, the reaction time of the user will likely be increased in real applications, stretching the relative performance measure Γ in the τ axis. However, as shown in Figure 7, for all cases, the influence of τ is negligible beyond approximately 10 times the expected user reaction time $\overline{t_r}$. Thus, if a sufficiently large $\tau > 10 \cdot \overline{t_r}$ is chosen, the performance of the algorithm will not be significantly impacted.

Moreover, with the proposed asynchronous access method, the user must only determine whether the device is behaving erroneously or not. In most cases, this should be obvious to them. Therefore, the actual reaction time may not be significantly longer than the simple visual reaction time considered in this experiment.

In terms of the second concern, the use of uncorrelated targets drawn from a uniform distribution has likely resulted in a lower boundary of performance for the experiments carried on here. In other words, the proposed method for asynchronous access is expected to have a better performance in a real application.

This is because real applications are likely to be composed of correlated targets whose spatial and temporal relationships are approximated by the basic assumptions incorporated in $\mathcal{X}$ and $\mathcal{H}$, respectively.

Finally, in cases where the user involuntarily rejects correct behaviors, (s)he will be forced to activate the interface a few more times in order to reach, once more, such behavior. However, it is important to note that, for the proposed example, the algorithm's performance will still be subject to the pattern reported in figure 6 even though the correct behavior will be placed at the end of the queue immediately after an involuntary rejection. That is, it will still take $X \simeq 18$ or less interface activations to reach the target again. On average, however, this process will take longer than with a random selection. Thus, for settings in which the probability of false-positives (i.e. involuntary rejections) is high, the performance of the algorithm may be significantly compromised. The reasons for increased false-positive rates in a specific application depend not only on the user's ability to maintain a particular selection, but also on the performance of the binary interface itself. In order to mitigate the incidence of false-positives, a variety of strategies can be used ranging from adaptations of the physical environment (including the interface) to the implementation of digital filters that disambiguate the user's intention. Due to the complexity and interdependence among the different factors that may influence performance in a specific context, this issue must be studied on a case-by-case basis. We will explore in detail the real impact of false-positives and its potential mitigation in further studies involving real users that attempt to control complex appliances using a binary interface in combination with the proposed algorithm.

From the results presented above, one may also observe that the number of parameters $\{\omega, \tau\}$ (i.e., the width of the spatial exclusion mask and the viscoelastic constant, respectively), which result in maximum relative performance Γ, increases with the tolerance σ of the particular application. Conversely, the maximum relative gain Γ, obtained with higher tolerances σ, is reduced in comparison to cases where the tolerance is small. Thus, for instance, while a wider range of parameters $\{\omega, \tau\}$ is acceptable in the case $\sigma = 0.2$, the maximum gain obtained with optimal parameters $\{\omega, \tau\}$ in the case $\sigma = 0.05$ is significantly higher. This phenomenon represents the main trade-off of the proposed method. Thus, in principle, the proposed asynchronous access method may be used to determine the behavior of a device with any degree of precision; however, higher precision will require more rigorous fine tuning of the algorithm parameters $\{\omega, \tau\}$.

In all cases, maximum values of performance were reached when $\omega = \sigma/2$. This actually corresponds to the cases where the exclusion mask $\mathcal{X}_{[n]}(c)$, characterized by ω, corresponded with the actual requirements of the application summarized by the tolerance σ. Evidently, if the tolerance σ of a particular application is known, the optimal value $\omega = \sigma/2$ may be immediately set. However, in a real application, this tolerance may not be easily identified. Furthermore, tolerance is likely to depend on the control priorities of each user in a particular application. Thus, for example, within the maximum tolerance for the execution of a particular task, some users may be more willing to accept errors than others. Nevertheless, the wide variety of arrangements available through the proposed access method, allows for its adaptation to virtually any type of user.

A sample application

In order to demonstrate the relevance of the proposed asynchronous access method in the control of real appliances, we implemented a single-switch drawing application termed the *one-button doodler* [21], this on-line tool allows minimal interface users to create "free-hand" drawings using only the left button of a computer mouse or a single keyboard key. Figure 8 depicts a sample drawing made by a minimal interface user by means of this software application. A particular implementation of the proposed method, with spatial exclusion mask $\mathcal{X}_{[n]}(c)$ defined by Equation (14), allowed this user to access a total of 180 different angles, allowing a pointer to follow the trajectory defined by the depicted trace. Note that, given the angular nature of the domain under control, the mask introduced here was more appropriate for this application than the mask defined in Equation (4) above.

$$\mathcal{X}_{[n]}(c) = \frac{1}{2}[1 + \cos(r)] \tag{14}$$

We have previously reported that, in order to minimize the delay between the user's action and the device's response, the outcome $c_{[n]}$ is transmitted to the device immediately after it is selected. However, in recent experiments involving real users, it has been evident that this immediacy is not as important as the ability to select the intended behaviour with high accuracy. Thus, in some cases, users prefer to have some time to reject the most recent selection proposed by the algorithm. Furthermore, since the algorithm becomes highly predictable soon after the interaction with the user has been initiated, it is possible to display a list of suggested behaviors that will follow

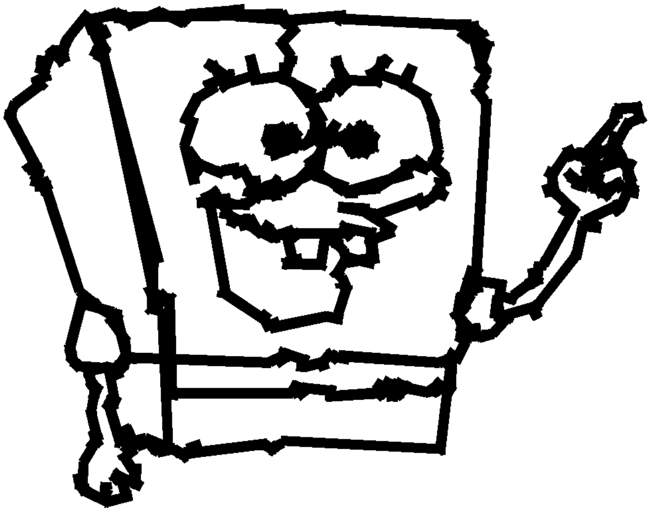


Figure 8

Sample drawing made by a minimal interface user by means of the one-button doodler on-line software application [21]. A particular implementation of the proposed method, with spatial exclusion mask $\mathcal{X}_{[n]}$ (c) defined by Equation (14), allowed this user to access a total of 180 different angles, allowing a pointer to follow the trajectory defined by the depicted trace.

the most recent selection. When available, this additional information can improve accuracy significantly. Full results of these and other studies involving real users will be reported in subsequent publications.

Conclusion

A novel method of asynchronous binary access has been proposed. This method translates consecutive intentional changes, executed by users of binary interfaces at irregular intervals, into increasingly accurate estimates of their intention. With this method, users are required to employ their interfaces only when the device under control behaves erroneously. When this happens, an algorithm that incorporates simple spatial and temporal assumptions, regarding all possible device outcomes, is used to obtain an informed estimate of the best possible outcome that the device should present next. This algorithm is based on the mechanical deformation of a viscoelastic space that may store the set of historical assumptions preceding any intentional event performed by the user. The theoretical evaluation of this method resulted in two significant conclusions:

1. The proposed method may be used with binary interfaces to asynchronously access devices with any number of potential outcomes and,

2. this method may be optimized through the particular choice of the spatial and temporal exclusion masks $\mathcal{X}$ and $\mathcal{H}$, according to the particular requirements and contextual circumstances of each application.

Authors' contributions

JS designed the asynchronous selection algorithm, the software tools and data structures for the experiments. JS proposed the initial design of experiments, executed them, and analyzed and interpreted the data. JS worked on the initial draft of the manuscript. JT, TC and AM advised upon the design and coordination of the study, experiments and data analysis, and multiple revisions of the manuscript. All authors read and approved the final version of the manuscript.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The authors would like to thank the support of the Health Care, Technology and Place interdisciplinary research program at the University of Toronto. Additional contributions from the Peterborough K. M. Hunter Foundation, the Toronto Rehabilitation Institute, the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Canadian Institutes for Health Research (CIHR), and the Bloorview Research Institute are also acknowledged.

The authors would also like to acknowledge the support and advice of Mr. Michael Dzura in the development of this work.

References

1. Cook AM, Hussey SM: *Assistive Technologies Principles and Practice* Mosby, Inc; 2002.
2. Shannon CE: **A mathematical theory of communication.** *Bell System Technical Journal* 1948, **27:**379-423.
3. Millan JR, Renkens F, Mourino J, Gerstner W: **Brain-actuated interaction.** *Artificial Intelligence* 2004, **159:**241-59.
4. Gage GJ, Ionides EL, Kipke DR: **Information capacity of brain machine interfaces.** *Proceedings of the 27th Conference of the IEEE Engineering in Medicine and Biology Society* 2005.
5. Lombardi J, Betke M: **A Self-initializing Eyebrow Tracker for Binary Switch Emulation.** *Tech Rep BUCS-TR-2002-023* 2002 [http://citeseer.ist.psu.edu/lombardi02selfinitializing.html]. Boston University, Computer Science Department
6. Grauman K, Betke M, Gips J, Bradski GR: **Communication via eye blinks – detection and duration analysis in real time.** *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2001, **1:**1010-7.
7. Mason SG, Birch GE: **A Brain-Controlled Switch for Asynchronous Control Applications.** *IEEE Transactions on Biomedical Engineering* 2000, **47(10):**1297-307.
8. Allanson J: **Electrophysiologically interactive computer systems.** *Computer* 2002:60-5.
9. Luo CH, Shih CH: **Adaptive Morse-coded single-switch communication system for the disabled.** *International Journal of Bio-Medical Computing* 1996, **41(2):**99-106.
10. Yang C, Chuang L, Yang C, Luo C: **Internet access for disabled persons using morse code.** *International journal of computers & applications* 2004, **26:**10-6.
11. Hauck LT: **SAM: An Improved Input Device.** *Proceedings of the Johns Hopkins National Search for Computing Applications to Assist Persons with Disabilities* 1992.

12. Blackstien-Adler S, Shein F, Quintal J, Birch S, Weiss PL: **Mouse manipulation through single-switch scanning.** *Assistive Technology* 2004, **16:**28-42.
13. Simpson R, Koester H, LoPresti E: **Evaluation of an adaptive row/column scanning system.** *Technology and Disability* 2006, **18:**127-38.
14. Shein GF: **Towards task transparency in alternative computer access: Selection of text through switch-based sacanning.** In *PhD thesis* University of Toronto; 1997.
15. Liu H, Ma H, Zarki ME, Gupta S: **Error control schemes for networks: An overview.** *Mobile Networks and Applications* 1997, **2(2):**167-81.
16. Veeraraghavan M, Wang H: **A Comparison of In-Band and Out-of-Band Transport Options for Signaling.** *Proceedings of the IEEE Communications Society Globecom Workshops* 2004:345-51.
17. Oppenheim AV, Willsky AS: *Signals & Systems* Prentice Hall; 1996.
18. Leon-Garcia A: *Probability and Random Processes for Electrical Engineering* second edition. Addison-Wesley Publishing Company, Inc; 1994.
19. Rubinstein RY: *Simulation and the Monte Carlo Method* New York, NY, USA: John Wiley & Sons, Inc; 1981.
20. Laming DRJ: *Information Theory of Choice-Reaction Times* London: Academic Press; 1968.
21. Wang S, Dzura M, Silva J: **One-Button Doodler.** [http://doodler.komodoopenlab.com].